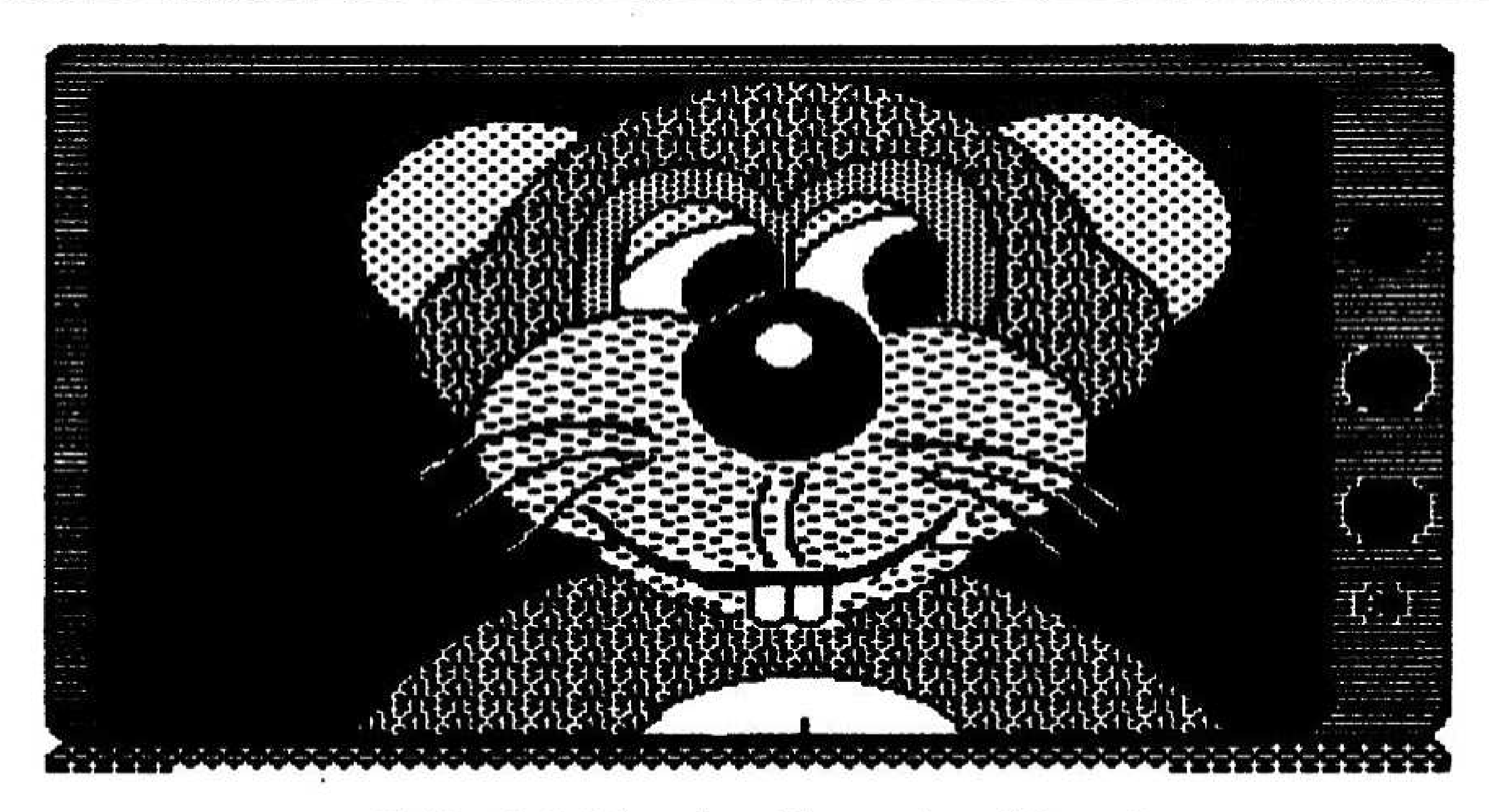
SPECTRUM PROFI CLUB

für Spectrum und SAM-User



Hallo, ich bin eine Kempston-Maus !

Inhalt:

Wolfgang Haller
Ernastr. 33
5000 Köln 80
Tel. 0221/685946

INFO
3/91

Smalltalk

Aus dem Club...

Durch den Eintritt dreier neuer Mitglieder erhöht sich die Mitgliederzahl auf 104. Ein herzliches Willkommen im SPC an:

Holger Dittmann, Oster Toft 12, 2396 Sterup
Christian Scharmberg, Olvenstetter Grund 27, O-3042 Magdeburg
Hartmut Sonntag, Borsigstr. 28, O-1040 Berlin

Und nun noch eine Adressenänderung:
Slawomir Grodkowski, Am Schlehdorn 6, 3400 Göttingen

Interessante Adressen:

Spectrum-Software wird noch vertrieben von:
Jupiter-Soft, Frühlingstr. 12
8831 Weiboldshausen

Hardware und Ersatzteile (bzw. eine Liste gegen 5 DM in Briefmarken) gibt es bei:
Decker & Computer, Postfach 100923, 7000 Stuttgart 10

Für Opus-Besitzer gibt es auch einen Opus-Discovery-Club:
SDC, B.D.Mumford, 57 St.Saviours Road, West Croydon, Surrey, CRO 2XE, England

Wer weitere interessante Adressen rund um den Spectrum oder Sam Coupe weiß, der schreibe mir bitte. Wir suchen vor allem Adressen von zuverlässigen (!!!) Leuten, die Spectrum und/oder Hardware reparieren können. Leider gab es gerade hier in letzter Zeit für manche Clubmitglieder teilweise Riesenärger.

Was noch?

Wer möchte gerne die Programm-Neuerscheinungen übernehmen? Was haltet ihr von einer Software Top 10? Wer hat Erfahrungen mit dem Z88 und teilt sie uns mit? Gibt es Neuigkeiten über den "Hobbit"?
Erfreulich: In letzter Zeit höre ich vermehrt von interessanten Aktivitäten rund um den Spectrum. So weiß ich zum Beispiel von Bemühungen, den Spectrum über einen ULA-Nachbau auf 10 Mhz ! zu takten. Ebenso ist es (mehreren) Mitgliedern gelungen, den 128er Spectrum nachzubauen. Auch die Opus wurde schon nachgebaut. Es würde mich freuen, wenn diese Leute ihr Wissen auch an andere interessierte User hier über die RU weitergeben würden. Wie wär's?
Danke für die eingesandten Demos von DMC und ZB of TMG. Beide sind sehr gut gelungen, beim einen der "Gag über den 64er" und beim anderen der hervorragende 128er Sound. Aber warum immer so ellenlange Texte (danke für die Grüße!)? Und was spricht gegen ein Demo in Basic? Es kommt doch auf die "Aussage" des Demos an, oder???
Und noch was in eigener Sache: Mancher Artikel muß etwas "zurückstehen", weil ich ihn aus Platzgründen nicht direkt unterbringen konnte. Meist erscheint er dann eine Ausgabe später. Also bitte nicht gleich mit mir schimpfen.

Denksportaufgabe

Vor kurzer Zeit stellte sich mir das Problem, das ich ein Tischfußballtrunier zu organisieren hatte. Das Problem besteht nun darin, daß ich aus einer vorgegebenen Zahl von Mannschaften alle Möglichkeiten (jeder spielt gegen jeden) brauche, und diese dann in einem Spielplan anordne. Z.B. 4 Mannschaften -> 6 Möglichkeiten:
1-2,1-3,1-4,2-3,2-4,3-4.
Das eigentliche Problem besteht nun darin, alle Möglichkeiten im Spielplan so anzuordnen, das es z.B. nicht vorkommt, das eine Mannschaft unmittelbar zweimal nacheinander spielen muß. Ferner sollte es so sein, das jede Mannschaft etwa die gleichen Pausenzeiten zwischen den eigenen Spielen hat, damit die Mannschaften gleichviel und gleichmäßig belastet werden.
Ich würde mich nun freuen, wenn Ihr Euch mal ein paar Gedanken über eine Lösung meines Problems machen und Eure Lösung dann als BASIC-Programm formulieren würdet. Damit das Ganze nicht ohne Ansporn geschieht, und ihr die Flinte nicht gleich ins Korn werft:

==
= Als Preis für die beste Lösung =
= gibt es: =
= Entweder 2-3 Spectrum Bücher oder =
= 2-3 Originalprogramme !!! =
==

Die Lösungen bitte hier im Clubheft veröffentlichen, oder mir auf Cassette, Beta-Disk bzw. als Listing zuschicken. DANKE.
Bernhard Lutz, Obermühlstraße 24 (bei Wünschel),
6729 Bellheim, Tel. 07272/6868

GM-BASE REVIEW

Dies ist das erste Database-Programm für den SAM-Coupe. Es ist komplett in Basic geschrieben und somit relativ einfach zu modifizieren. Es benötigt 256K Hauptspeicher und ein Diskettenlaufwerk.

Das Programm wird in zwei Teilen geladen und nach einem hübschen Ladescreen wird das "Main Menu" gezeigt:

CREATE DISK EDIT PRINT

Eine Option kann nun mit den Cursortasten selektiert werden. Es ist empfehlenswert, jetzt eine "Backup"-Kopie zu machen, bevor man anfängt, zu arbeiten. Als erstes selektiert man "DISK" und drückt "Return". Es wird ein neues Menu angezeigt:

LOAD SAVE OTHER MENU

Nachdem wir "Other" angewählt haben, wird das "File Manipulations Menu" gezeigt:

RENAME COPY ERASE MENU

Selektiert man nun "Copy", braucht man nur noch den Befehlen am Bildschirm zu folgen, und auf eine vorformatierte Diskette wird eine Backup-Kopie geschrieben. Zurück zum "Main Menu" und ein database kann generiert werden. Wählt "create" und den Typ des gewünschten database-Namens. Ein database-Namen darf nicht mehr als 7 Buchstaben haben, weil es automatisch mit ".db" ergänzt wird, damit es als database-File identifiziert werden kann.

Nach "Return" zeigt der Bildschirm 10 Zeilen an. In diese kann man bis zu 10 Titel für Felder definieren (z.B. Nachname, Vorname usw.). Jedes Feld kann zwischen 3 und 20 Buchstaben beinhalten. Wenn man fertig ist, beendet "Quit" diese Phase, und das database ist für die Dateneingabe vorbereitet. Mit "Edit" kann man die Dateneingabe vornehmen. Es steht jederzeit eine "HELP"-Funktion zur Verfügung, man muß nur die "Edit"-Taste drücken. Es gibt 14 Edit-Möglichkeiten mit "delete a record", "sort records" usw. Es ist auch möglich, individuelle Records auszudrucken, entweder tabuliert oder in einem freien Format.

Wie gut ist GM-BASE? Nicht schlecht, die Unterlagen sind leicht verständlich (wenn man Englisch kann), die "Help"-Seite auf dem Bildschirm ist sehr nützlich. Der "Sort" ist ziemlich schnell und sehr gut durchdacht. Es genügt die Eingabe einiger Buchstaben um das entsprechende Record zu holen. Restriktiv ist folgendes: Es kann zur Zeit nur 249 Records verwalten, das ist wirklich nicht viel, mit "VU-FILE" unter "Specmaker" kann man ein ähnlichs "test-file" mit über 800 Records generieren. Aber wenn man nicht so viele braucht bzw. wenn die 249 Records reichen, dann hat man mit GM-BASE ein gut durchdachtes Programm.

Ein zweites Problem ergibt sich, wenn man durch einen Error ins Basic gelangt. s ist nicht leicht, wieder ohne Datenverlust ins Programm einzusteigen. Aus diesem Grund sollte man notwendigerweise öfters seine Daten auf Diskette sichern.

Laut Information der Entwickler arbeiten sie zur Zeit an einer Version in Maschinencode, etwas was sicher notwendig ist, wenn man mehr als 249 Records verwalten will.

So - wo bekommt man dieses Programm? Die Adresse ist:
GM SOFTWARE - 48, Main Road, Crynant, Neath SA10 8NP, ENGLAND
und der Preis in England 4,95 Pfund (15,- DM).

Dieses Review basiert auf der Erfahrung von Frank Harrop, einem Freund von mir, und wurde von mir frei übersetzt.

Ian D. Spencer, Fichtenweg 10c, 5203 Much, Tel. (02245) 1657

Die Opus Discovery, Teil 9

Nachdem wir das letzte mal mit der Opus Programme geladen und gespeichert haben, wollen wir uns heute der Vollständigkeit wegen mit den restlichen Routinen der ersten Tabelle beschäftigen.

CLEACH: Diese Routine funktioniert genauso wie CLOSCH (RU 2/91). Der einzige Unterschied ist, daß der Kanal vor dem Entfernen aus dem Speicher nicht geschlossen wird. Sie wird auch genauso wie CLOSCH aufgerufen. (CLOSCH entspricht z.B. dem BASIC-Befehl CLOSE # 4; CLEACH entspricht CLEAR # 4)

WIPECH: Diese Routine entspricht dem BASIC-Befehl CLEAR # (entfernen aller Kanäle aus dem Speicher). Aufgerufen wird die Routine einfach mit CALL WIPECH.

MOVECH: Hiermit kann man wie im BASIC-Befehl MOVE Daten von einem Eingabekanal auf einen Ausgabekanal bewegen. Vorher müssen die beiden Kanäle natürlich mittels OPENCH eröffnet werden (zur Erinnerung: Anfangsadresse steht nach Aufruf von OPENCH in IX). Dann wird MOVECH wie folgt aufgerufen:

```
LD HL,<Anfangsadresse des Eingabekanals>
LD IX,<Anfangsadresse des Ausgabekanals>
CALL MOVECH
```

Jetzt werden alle Daten aus dem Eigabekanal zum Ausgabekanal bewegt, bis im Eingabekanal eine Endmarke erreicht wird.

CALCHN: Als letzte Routine der ersten Tabelle kommt nun die CALCHN-Routine. Mit dieser Routine können wir eine Menge Funktionen mit der Opus ausüben, die wir zum Teil auch schon mit den vorher genannten Routinen tun konnten. Vor dem Aufruf von CALCHN muß in IX wieder die Anfangsadresse des Kanals stehen (mit OPENCH zu ermitteln!). Im B-Register steht eine Zahl, die dem Speccy signalisiert, welche Funktion wir mit CALCHN ausüben wollen. So ergibt sich folgender Aufruf dieser Routine:

```
LD IX,<Kanalanfang>
LD B,<Nr. der Unterroutine>
wenn nötig müssen hier noch andere Register mit Werten geladen werden
(welche Register steht im Folgenden)
CALL CALCHN
```

Schreiben in Kanal; LD B,0 (00h):
Mit dieser Funktion kann man das im A-Register enthaltene Zeichen in den Kanal schreiben.

Lese aus dem Kanal; LD B,2 (02h):
Hiermit ist es möglich Zeichen aus dem geöffneten Kanal zu lesen. Nach dem Einlesen setzt der Spectrum das Carry-Flag. Das Zeichen kann nun aus dem A-Register gelesen werden. Ist das Carry-Flag nicht gesetzt (wurde also kein Zeichen gelesen), dann spielt das Zero-Flag eine Rolle: Ist das Zero-Flag nicht gesetzt, dann wird damit angezeigt, daß der Kanal am Ende angelangt ist und kein weiteres Zeichen gelesen werden kann. Ist das Zero-Flag gesetzt, dann wird damit angezeigt, daß der Kanal noch nicht zu Ende ist, das neue Zeichen aber noch nicht gelesen werden konnte (der Speccy ist eben sehr viel schneller als das Laufwerk; damit es hier nicht zu Konflikten kommt kann man mit dem Zero-Flag überprüfen, ob das Laufwerk schon so weit ist)

Öffne Kanal; LD B,4 (04h):
Diese Funktion kann eigentlich gar nicht benutzt werden, da man zum Aufruf den Anfang des Kanals braucht (in IX). Diese Adresse hat man aber erst nach dem Aufruf. Man benutzt zum öffnen also besser OPENCH.

Schliesse Kanal; LD B,6 (06h):
Diese Funktion ist verwandt mit CLOSCH und CLEACH. Man kann hiermit einen

Kanal schliessen. Er wird jedoch danach nicht aus dem Speicher entfernt (CLOSCH entfernt und schließt; CLEACH entfernt nur und schließt nicht).

Länge ermittlen; LD B,8 (08h):
Nach Aufruf dieser Funktion hat man die Länge des aktuellen Kanals suf dem Kalkulatorstapel und kann ihn von dort dann herunterlesen.

Positioniere Kanal; LD B,10 (0Ah):
Diese Funktion ist identisch mit der POINT #-Funktion in BASIC. Vor dem Aufruf muß die gewünschte Position auf den Kalkulatorstapel abgelegt werden, so daß sich folgender Aufruf ergibt:

```
LD BC,<Position>
RST 16 ;=10h ; Aufruf im Speccy-ROM
DEFW 11563; =2D2Bh ; BC wird auf Kalkulatorstapel gelegt
LD IX,<Anfangsadresse>
LD B,10 ;=0Ah
CALL CALCHN
```

Zum Schluß möchte ich noch ein Lob an Scott-Falk Hühn los werden, der endlich einmal eine Serie über den Soundchip vom 128er Speccy schreibt. Auch seine Sounddemos, die er geschrieben hat sind wirklich erstklassig (man wundert sich, was alles in BASIC damit möglich ist).

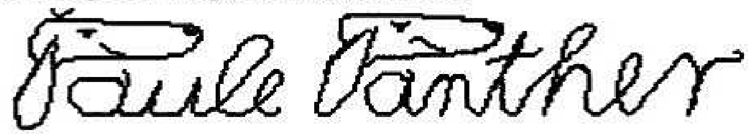
So, das war's wieder einmal für heute, bis bald,

Rüdiger Döring, Meisenstraße 10, 5467 Vettelschoß, Tel. 02645/3060

Spectrum-Geister ?



Eine etwas merkwürdige Frage möchte ich Euch heute stellen;
Was hat das rauchen mit dem Speccy zu tun ? Nein, ich meine nicht wenn der Speccy raucht. Dann ist eh alles zu spät. Nein, ich meine diese ungesunden Lungenbrötchen. Nichts, werdet Ihr wohl sagen. Und es stimmt auch im Grunde. Aber ein seltsamer Fehler in meiner Anlage hat mich auf diese Frage gebracht.
Erst habe ich beim auftreten dieses Fehlers an Geister geglaubt. Dann an einen Wackelkontakt. An Netzspannungs Einbrüche... aber der Reihe nach von vorn.
Ich sitze wie so oft am Speccy und will einen Brief schreiben. An und für sich ist mein Speccy ziemlich absturzsicher und Wackelkontaktfrei. Kann also Briefe ohne ständige Sicherheitscopien schreiben. Der Brief ist fertig und ich lasse ihn ausdrucken. Plötzlich SCH...EIBENKLEISTER! Bunte Quadrate auf dem Screen und der Drucker dreht durch! Na ja, kann ja mal vorkommen. War wohl ein Spannungseinbruch. Peinlich nur, ich muss den ganzen Brief neu schreiben, weil nicht gesichert.
Brief also nochmal geschrieben und gesichert. Ausdrucken und ?????? schon wieder Absturz! Nanü? Erst mal alle Stecker am Bus neu einwackeln. Noch'n Versuch. Aha! Geht. War wohl doch was lose. Brief eingetütet und erst mal ein Lungenbrötchen auf den Schreck. Aber was ist das??? nun hat sich das Multiface von selber eingeschaltet! Grübel Grübel und studier ???? Keine Ahnung was das sein soll. Noch nie dagewesen.
Plötzlich setzt sich das Disclaufwerk von selber in Bewegung. HILFE.. GEISTER! Genervt gehe ich für heute auf.
Am nächsten Tag das gleiche Spiel. Bunte Quadrate, Multiface kommt von selbst, Drive läuft an. Aus Verzweiflung den Ersatzspeccy rausgekramt und angeworfen. Aber auch hier das selbe. So langsam glaube ich wirklich an Gespenster!
Und dann..... ich schütte mir eine Tasse Kaffe in den Hals und lege eine Camel obendrauf und ????? HÄH ???? Beim anzünden des Feuerzeuges schaltet sich das Multiface ein! Ei Donnilofski!
Noch'n Versuch. Drive läuft an. Hihihihi! Was für'n Sauhund hat mir da ein Antiraucher Virus in den Speccy geschmuggelt??
Und dann die Erleuchtung! Bei dem Feuerzeug handelt es sich um ein Piezo Zünder!! Je nach dem wie es gerade beim zünden auf den Speccy zeigt, verursacht es die merkwürdigsten Fehler!
Und die Moral von der Geschicht', der Speccy mag das Rauchen nicht!



Spectrum 128 und Musik (Teil 2)

Hallo Musik-Freunde!

Ich möchte nun die Beschreibung der String-Kommandos im PLAY-Befehl des 128er-Basic fortsetzen.

Gleich zu Beginn möchte ich noch etwas zum "O"-Kommando ergänzen: Die Oktave 0 und die Noten c-gis in Oktave 1 können vom PSG nicht erzeugt werden, sie haben nur im Zusammenhang mit MIDI Bedeutung. Das Oktav-Kommando gilt für alle folgenden Noten, bis durch ein anderes "O" ein neuer Oktavbereich festgelegt wird. Günstige Oktaven für Melodie-Stimmen sind O4-O6, für Baßstimmen O2 und O3. Wird im Notenstring die Oktavangabe weggelassen, so gilt O5.
 Wenn nur einzelne Noten den eingestellten Oktavbereich überschreiten, so muß nicht unbedingt mit dem "O"-Kommando eine neue Oktave eingestellt werden. Man kann auch mit den Zeichen "#" und "$" die Grenznoten weiter erhöhen oder vermindern. Z.B. läßt sich aus O5 durch "###B" der Ton d in O7 erreichen (letzter Ton H wird um 3 Halbtonschritte erhöht) oder mit "$$$c" läßt sich der Ton a in O4 spielen.
 Ein weiteres wichtiges Kommando ist das "N"-Kommando. Es hat zwar keinen Einfluß auf die Tonerzeugung, ist aber für den Stringaufbau unentbehrlich. Folgendes Beispiel soll das verdeutlichen: es soll die Achtelnote f in Oktave 4 erzeugt werden, der String muß dann so aussehen: "O4N3f". Ohne das "N" würde der sinnlose Oktavwert 43 gelesen und das Programm mit einer Fehlermeldung abbrechen. Das "N" ist also überall dort, wo verschiedene Zahlen direkt nebeneinander stehen, als Trennzeichen zu verwenden.
 Mit den bisher beschriebenen Kommandos kann man schon kleine 3-stimmige Musikstücke programmieren. Dabei wird man feststellen, daß sich Teile einer Stimme, z.B. die Baßstimmen in der Pop-Musik, oftmals in bestimmten Abständen wiederholen. Hier kann man durch Setzen von Klammern "(" und ")" Teile eines Strings oder den gesamten String mehrmals spielen lassen. Alle Noten zwischen jeweils einer öffnenden und einer schließenden Klammer werden dann zweimal gespielt, bei jeder weiteren Einklammerung verdoppelt sich die Schleifenzahl. Es können maximal 5 Klammern gleichzeitig geöffnet werden (entspricht 32 Schleifen).
 Wird an das Ende des Strings nur eine schließende Klammer angefügt, so wird der gesamte String endlos wiederholt. Der gleiche Effekt wird erreicht, wenn mehr schließende als öffnende Klammern vorhanden sind. Hiermit kann man sehr einfach ständig wiederkehrende Baßläufe oder Backgrounds programmieren. Aber wie kann man diese Endlosschleifen am Ende des Musikstückes wieder verlassen? Auch für diesen Fall ist gesorgt. Voraussetzung ist, daß wenigstens eine Stimme (meist ist dies die Melodiestimme) nicht endlos ist. An das Ende dieses Strings wird dann der Buchstabe "H" angefügt. Dieses Kommando bewirkt beim Spielen des Strings an dieser Stelle den sofortigen Abbruch aller Endlosschleifen.
 Ich möchte nun ein weiteres Kommando beschreiben und wer schon einmal eine Melodie mit BEEP-Befehlen geschrieben hat, wird dieses sehr zu schätzen wissen, denn wenn das Tempo nicht stimmt, so muß man jeden einzelnen Ton verändern bzw. sehr aufwendig mit Variablen arbeiten. Bei PLAY ist dies nicht nötig, denn hier kann man am Anfang des Strings mit Hilfe des "T"-Kommandos das Tempo festlegen. Dem "T" muß eine Zahl zwischen zwischen 60 und 240 folgen. Diese gibt das zu spielende Tempo in Viertelnoten pro Minute an. Wird die Tempoeinstellung weggelassen, so spielt PLAY mit 120 Viertelnoten pro Minute. Man kann also das Tempo eines solchen Musikstückes in kleinen Schritten bis zur Hälfte herabsetzen oder bis zur Verdoppelung erhöhen. Das Tempo kann auch jederzeit an beliebiger Stelle im Notenstring geändert werden, die Tempoeinstellung sollte jedoch immer im ersten String erfolgen, sonst wird sie ignoriert. Das gewählte Tempo gilt dann selbstverständlich für alle Strings gleichermaßen.
 Außer dem Notenwert und der Notendauer hat jeder Ton in einem Musikstück noch eine andere Eigenschaft: die Lautstärke. Fast jedes Musikwerk ist dynamisch, d.h. es hat leise und laute Stellen, die den Gesamteindruck mitprägen. Auch solche Lautstärkeänderungen können mit dem PLAY-Befehl programmiert werden. Dazu

wird der Buchstabe "V", gefolgt von einer Zahl zwischen 0 und 15, verwendet. Die Lautstärke kann also von 0 (Ton ausgeschalten) bis 15 (volle Lautstärke) in kleinen Stufen programmiert werden.
Nimmt man eine Durchschnittslautstärke von 12 an, so kann man mit schrittweiser Erhöhung auf 15 eine bestimmte Stelle akustisch hervorheben, oder bei entsprechender Absenkung eine "Erholungspause" einbauen. Weiterhin kann man bei schrittweisem Anstieg von 0 auf 15 das Musikstück langsam einblenden oder in umgekehrter Reihenfolge wieder ausblenden. Wird kein "V"-Kommando benutzt, so wird der gesamte Notenstring in voller Lautstärke gespielt (entspricht V15). Die Lautstärkeeinstellung gilt nur für einen String, man kann somit die Lautstärke für jeden Notenstring gestrennt einstellen. Das hat den Vorteil, daß man z.B. eine Backgroundstimme etwas leiser spielen kann, um die Melodie besser zur Geltung zu bringen oder die Baßstimme in Melodiepausen etwas anheben kann oder..

Im 1. Teil habe ich bereits erwähnt, daß man mit dem PSG auch Geräusche erzeugen kann. Dazu möchte ich noch einmal in Erinnerung rufen, daß der PSG drei getrennte Soundkanäle besitzt. Jeder einzelne Kanal kann nun wahlweise einen Ton, ein Rauschsignal oder beides gleichzeitig erzeugen. Dazu besitzt der PSG ein Mixtur-Register, in dem die Zuordnung eingestellt werden kann. Dieses Register wird ebenfalls vom PLAY-Befehl unterstützt. Dazu dient der Buchstabe "M", gefolgt von einer Kennzahl zwischen 0 und 63. Diese Kennzahl setzt sich aus der Wertigkeit von 6 Bits zusammen, wobei diese Bits folgende Bedeutung haben:

Rauschkanal	C	B	A
Bit	D5	D4	D3
Wert	32	16	8

Tonkanal	C	B	A
Bit	D2	D1	D0
Wert	4	2	1

Die Verwendung des "M"-Kommandos läßt sich am besten an einem Beispiel zeigen: Es soll auf Kanal B ein Rauschen, auf den Kanälen A und C Töne erzeugt werden. Man addiert nun die Werte der benötigten Bits: 16+4+1 = 21. Die gewünschte Kanalbelegung wird durch "M21" eingestellt. Wird kein "M"-Kommando benutzt, so wird "M7" als Standart eingestellt, d.h. alle 3 Kanäle erzeugen nur Töne.

Hier noch einige Hinweise zum "M"-Kommando:

- Vom PLAY-Befehl wird der erste Notenstring immer dem Soundkanal A zugeordnet, der zweite dem Sounkanal B usw.
- Wird ein Kanal zur Gräuscherzeugung herangezogen, so wird anstelle einer Note ein Rauschsignal erzeugt, welches in der Tonlänge und Lautstärke dem Ton äquivalent ist, bei Verwendung von Kanal A entspricht das erzeugte Rauschspektrum sogar der Tonhöhe, d.h. bei tiefen Noten erklingt ein "dunkles", bei hohen Noten ein "helles" Rauschen.
- Der PSG besitzt intern eigentlich nur einen Rauschgenerator, es ist deshalb wenig sinnvoll, mehrere Kanäle gleichzeitig für Geräusche zu verwenden.
- Kommandos wie "M0" oder "M63" sind prinzipiell möglich, liefern aber kaum brauchbare Ergebnisse.
- Eine optimale Geräuscherzeugung ist nur mit Kanal A möglich, hier kann durch unterschiedliche Notenwerte der Charakter des Rauschens beeinflußt werden. In Musikstücken kann man damit verschiedene Schlagzeugeffekte einspielen.
- Das "M"-Kommando kann jederzeit in einem beliebigen String eingesetzt werden, um eine neue Kanalbelegung festzulegen, diese wirkt dann entsprechend auf alle Kanäle. Dadurch kann man z.B. Kanal A gleichzeitig für Baßstimme und Schlageffekt nutzen, indem man an geeigneter Stelle die Kanäle umschaltet ("M7" für Baßstimme, "M14" für Effekte).

An dieser Stelle möchte ich den heutigen Beitrag abschließen. Beim nächsten Mal kommen die restlichen Kommandos und MIDI an die Reihe. Tschüß bis bald

Scott-Falk Hühn, Erich-Heyl-Str. 4, Sömmerda / Thüringen, O-5230
Tel.(O): 00626 22467 / Tel.(W): 0037 626 22467

Tag allerseits!!

Der Befehlssatz des Zilog Z 80 / Teil 8

LD R,A
Lade das Memory-Refresh-Register aus dem Akkumulator.
Der Inhalt des Akkumulators wird in das Memory-Refresh-Register geladen.
Beispiel: *LD R,A*

LD r,(HL)
Lade das Register r indirekt aus der Speicherstelle (HL).
Der Inhalt der Speicherstelle, die durch HL adressiert wird, wird in das angegebene Register geladen.
Beispiel: *LD D,(HL)*

LD SP,HL
Lade den Stapelzeiger aus HL.
Der Inhalt des Registerpaares HL wird in den Stapelzeiger geladen.
Beispiel: *LD SP,HL*

LD SP,IX
Lade den Stapelzeiger aus dem Register IX.
Der Inhalt des Registers IX wird in den Stapelzeiger geladen.
Beispiel: *LD SP,IX*

LD SP,IY
Lade den Stapelzeiger aus dem Register IY.
Der Inhalt des Registers IY wird in den Stapelzeiger geladen.
Beispiel: *LD SP,IY*

LDD
Blockladen mit Dekrementieren.
Der Inhalt der Speicherstelle, die durch HL adressiert wird, wird in die Speicherstelle geladen, auf die DE zeigt. Dann werden DE, BC und HL dekrementiert.
Beispiel: *LDD*

LDDR
Wiederholtes Blockladen mit Dekrementieren.
Der Inhalt der Speicherstelle, die durch HL adressiert wird, wird in die Speicherstelle geladen, auf die DE zeigt. Dann werden DE, BC und HL dekrementiert. Ist BC ungleich Null, dann wird der Befehlszähler um 2 dekrementiert und der Befehl wiederholt.
Beispiel: *LDDR*

LDI
Blockladen mit Inkrementieren.
Der Inhalt der Speicherstelle, die durch HL adressiert wird, wird in die Speicherstelle geladen, auf die DE zeigt. Dann werden DE und HL inkrementiert und BC dekrementiert.
Beispiel: *LDI*

LDIR
Wiederholtes Blockladen mit Inkrementieren.
Der Inhalt der Speicherstelle, die durch HL adressiert wird, wird in die Speicherstelle geladen, auf die DE zeigt. Dann werden DE und HL inkrementiert und BC dekrementiert. Ist BC ungleich Null, dann wird der Befehlszähler um 2 dekrementiert und der Befehl wiederholt.
Beispiel: *LDIR*

NEG
Negiere Akkumulator.
Der Inhalt des Akkumulators wird von Null subtrahiert (Zweierkomplement). Das

Ergebnis wird wieder im Akkumulator gespeichert.
Beispiel: *NEG*

NOP
Keine Operation.
Einen M-Zyklus lang wird nichts getan.
Beispiel: *NOP*

OR s
Logische Oder-Verknüpfung von Akkumulator und Operand s.
Der Akkumulator und der angegebene Operand s werden logisch durch die Funktion ODER verknüpft und das Ergebnis wieder im Akkumulator abgelegt.
Beispiel: *OR B*

OTDR
Blockausgabe mit Dekrementieren.
Der Inhalt der Speicherstelle, die durch das Registerpaar HL adressiert wird, wird zu dem peripheren Gerät ausgegeben, das durch den Inhalt des Registers C adressiert wird. Dann werden die Register B und HL dekrementiert. Ist B ungleich Null, dann wird der Befehlszähler um 2 dekrementiert und der Befehl nochmals ausgeführt. C liefert die Bits A0 bis A7 des Adressbusses, B liefert A8 bis A15 (nach dem Dekrementieren).
Beispiel: *OTDR*

OTIR
Blockausgabe mit Inkrementieren.
Der Inhalt der Speicherstelle, die durch das Registerpaar HL adressiert wird, wird zu dem peripheren Gerät ausgegeben, das durch den Inhalt des Registers C adressiert wird. Dann werden das Register B dekrementiert und HL inkrementiert. Ist B ungleich Null, dann wird der Befehlszähler um 2 dekrementiert und der Befehl nochmals ausgeführt. C liefert die Bits A0 bis A7 des Adressbusses, B liefert A8 bis A15 (nach dem Dekrementieren).
Beispiel: *OTIR*

OUT C,r
Ausgabe des Registers r zum Port C.
Der Inhalt des angegebenen Registers wird an das periphere Gerät ausgegeben, das durch den Inhalt des Registers C adressiert wird.
Beispiel: *OUT (C),B*

OUT (N),A
Ausgabe des Akkumulators an den peripheren Port N.
Der Inhalt des Akkumulators wird an das periphere Gerät ausgegeben, das durch den Inhalt der Speicherzelle adressiert wird, die unmittelbar auf den Op-Code folgt.
Beispiel: *OUT (0A),A*

OUTD
Ausgabe mit Dekrementieren.
Der Inhalt der Speicherzelle, die durch das Registerpaar HL adressiert wird, wird an das periphere Gerät ausgegeben, das durch den Inhalt des Registers C adressiert wird. Dann werden die Register B und HL dekrementiert. C liefert die Bits A0 bis A7 des Adressbusses, B liefert A8 bis A15 (nach dem Dekrementieren).
Beispil: *OUTD*

OUTI
Ausgabe mit Inkrementieren.
Der Inhalt der Speicherzelle, die durch das Registerpaar HL adressiert wird, wird an das periphere Gerät ausgegeben, das durch den Inhalt des Registers C adressiert wird. Dann wird das Register B dekrementiert und HL inkrementiert. C liefert die Bits A0 bis A7 des Adressbusses, B liefert A8 bis A15 (nach dem Dekrementieren).
Beispiel: *OUTI*

POP qq

Hole das Registerpaar qq vom Stapel.
Der Inhalt der Speicherzelle, auf die der Stapelzeiger zeigt, wird in die untere Hälfte des angegebenen Registerpaares geladen. Dann wird der Stapelzeiger inkrementiert und der Inhalt der Spicherzelle, auf die er dann zeigt, in die obere Hälfte des angegebenen Registerpaares geladen. Daraufhin wird der Stapelzeiger nochmals inkrementiert.
Beispiel: *POP BC*

POP IX

Hole das Register IX vom Stapel.
Der Inhalt der Speicherzelle, auf die der Stapelzeiger zeigt, wird in die untere Hälfte des Registers IX geladen. Dann wird der Stapelzeiger inkrementiert und der Inhalt der Speicherzelle, auf die er dann zeigt, in die obere Hälfte des Registers IX geladen. Daraufhin wird der Stapelzeiger nochmals inkrementiert.
Beispiel: *POP IX*

POP IY

Hole das Register IY vom Stapel.
Der Inhalt der Speicherzelle, auf die der Stapelzeiger zeigt, wird in die untere Hälfte des Registers IY geladen. Dann wird der Stapelzeiger inkrementiert und der Inhalt der Speicherzelle, auf die er dann zeigt, in die obere Hälfte des Registers IY geladen. Daraufhin wird der Stapelzeiger nochmals inkrementiert.
Beispiel: *POP IY*

PUSH qq

Lege das Registerpaar qq auf dem Stapel ab.
Der Stapelzeiger wird dekrementiert und der Inhalt der oberen Hälfte des angegebenen Registerpaares in die Speicherzelle geladen, auf die der Stapelzeiger zeigt. Dann wird der Stapelzeiger nochmals dekrementiert und der Inhalt der unteren Hälfte des Registerpaares in die Speicherzelle geladen, auf die der Stapelzeiger jetzt zeigt.
Beispiel: *PUSH DE*

PUSH IX

Lege das Register IX auf dem Stapel ab.
Der Stapelzeiger wird dekrementiert und der Inhalt der oberen Hälfte des Registers IX in die Speicherzelle geladen, auf die der Stapelzeiger zeigt. Dann wird der Stapelzeiger nochmals dekrementiert und der Inhalt der unteren Hälfte des Registers IX in die Speicherzelle geladen, auf die der Stapelzeiger jetzt zeigt.
Beispiel: *PUSH IX*

Bis demnächst hier im Info...

Harald R. Lack, Heidenauer Str. 5, 8201 Raubling

VORSTELLUNG

Liebe Clubfreunde!

Möchte mich hiermit vorstellen, mein Name ist Ernst Eulenbach, Hahnenbergerstr. 243, 5600 Wuppertal 12. Ich bin 70 Jahre. Durch eine schwere Krankheit (Schlaganfall) mußte ich meinen Beruf aufgeben, zu allem Übel wurde ich noch Diabetiker. Durch diese Krankheit bin ich zu einem ZX Spectrum 16K gekommen, um wieder denken zu lernen.
Zur Zeit habe ich folgende Geräte: 1 Spectrum 128K, 1 Spectrum 48K, 3 Microdrive, 1 Drucker Seikosha SP 1000 AS, 1 Drucker GP 50S und 1 Opus-Discovery 180/720K. Mein Spezial-Hobby ist: Foto (Landschaft, Architektur) sowie Macro- und Micro-Aufnahmen.
Nun habe ich noch eine Bitte, wer kann mir helfen? Ich suche seit langer Zeit für den 48K eine Schaltfolie, sowie das Würfelspiel Chicago. Ich würde mich freuen, wenn einer mir dies besorgen könnte.

Benchmark Tests (Vergleiche)

Betreff das Info 8/90 mit dem Compiler. Nun habe ich folgendes gemacht. Ich hab mal meinen alten ZX81 ausgegraben und ihn mal ausgetestet. 2 Stunden habe ich gebraucht, um den MC-Coder reinzuladen. Und 20 Minuten zum Testen. Auch habe ich für meinen Atari einen Spectrum- Emulator!!! Das Bild ist einfach SUPER!! Nur noch etwas langsam! Der Emulator ist eh eine Demo-Version. Spectrum-Emulator sind unter Tel. 0621/668078 zu haben. Die Programmierer testen gerade den Markt! Bitte mal dort zu tausenden anrufen!
Nun die Auswertungen meiner Testerei, ab Seite 60. Anscheinend haben es noch nicht alle User mitbekommen? Bei einem 80 Track-Laufwerk braucht man 2DD oder DDDD-Disketten! Also 96 TPI wie die HD-Disketten. Wer sagt da, daß ich teure HD-Disketten verheize??? Die 2DD (96 TPI) kosten 36 DM (10 Stück) und die HD-Disketten nur 22 DM (BASF). Anscheinend werden die 4D-Disketten nicht mehr hergestellt?? Bei Lerche (neue Zweigstelle in Stuttgart) konnte ich aus 4 Kartons die letzten 4D-Disketten ergattern. Das letzte und einzigste Pack kostete 9.90 DM (Maxell MD1-DD 96 TPI!!). Die 4D-Disketten gibt es nur für 5 1/4 Zoll Laufwerke! Bei den normalen 2D-Disketten habe ich nun ab- und zumal Probleme. Allerdings mit meinen HD-Disketten hat ein Spectrum-Kollege auch seine Probleme! Werde das Problem mal analysieren. Noch schlimmer ist es, wenn man ein Billig-Laufwerk hat (wie ich)! Beim Testen wurden die Benchmark-Tests 1-4, auf S. 4 der 8/90er Ausgabe übernommen. Zeile 10 wurde to 500 ab und zu verändert!!

	Benchmark-Test	1	2	3	4
Original Spectrum!	to 500	170,8	24,9	4,5	4,2
Spectrum-Emulator					
für Atari 1040 ST	to 50 !! leider !				
Mode F1		74,34			
Mode F2		142			
Mode F5	= verschiedene	91,3			
Mode F6	Grafik-	94			
Mode F7	Auflösungen	89			
Patches - ON		71			
Mode F7	mit to 500!!	---	126,1	19,1	15,5

Der Spectrum-Emulator ist ca. 4,157 bis 4,352 bis 8,314 mal langsamer.

ZX 81 Original!!					
FAST-Modus	to 500	164,5	23,7	4,44	3,85
MC-CODER SLOW	to 5000 !	24,3	NOT	7,8	----
MC-CODER FAST		6,6	NOT	2,3	----
MC-CODER SLOW	to 20000 !!				4,80
MC-CODER FAST					1,60
	rechnerisch bei to 500				0,12
					0,0405

Beim SLOW-Modus ist mein ZX-81 4,0094 mal langsamer als im FAST-Modus. Auch ist der Spectrum um den Faktor 1,08 langsamer, als der gute alte ZX-81! Wer hätte gedacht, daß der Spectrum langsamer ist? Und ist sogar schneller als der Atari mit GFA-Basic 2.00 mit Compiler!! Genau 0,9 mal!
Habe da noch den USCHI-Compiler gefunden (Spectrum 48K):

USCHI-Compiler	to 500	166,75	NOT	0,2	0,05
ToBoS	to 500	2,6	1,6	0,48	0,5
Einen ZX-81 Emulator					
habe ich auch noch ausgegraben			31,1??	18,0	13,8
ZX-81 Atari!!	to 500	-> nach 9 Min. abgebrochen <-			
FAST	to 50	77,7	2,0		
SLOW	to 50	93,6			
Beim Atari ST 1040 selber		0,9			

Anscheinend sind die angegebenen 3 Sekunden mit dem alten TOS gemessen worden?

Zuallerletzt wurden auch noch GFA-Basic-Versionen nach Benchmark Test 1 bis 4 ausgetestet. Und statt FOR-NEXT noch mit Repeat, While und Do ausgetestet. System 1040 ST mit altem TOS. Das sieht dann so aus:

	FOR	Repeat	While	Do
GFA-Basic 2.0 - 11stellig				
Interpreter	1= 1,39 2= 0,645 3= 0,17 4= 0,15	1= 1,235 2= 0,725 3= 0,255 4= 0,23	1= 1,25 2= 0,74 3= 0,265 4= 0,24	1= 1,25 2= 0,735 3= 0,265 4= 0,245
Compiler	1= 1,005 2= 0,515 3= 0,08 4= 0,045	1= 1,015 2= 0,54 3= 0,085 4= 0,055	1= 1,015 2= 0,53 3= 0,086 4= 0,06	1= 1,015 2= 0,53 3= 0,085 4= 0,06
GFA-Basic 3.0 - 14stellig				
Interpreter	1= 0,53 2= 0,725 3= 0,095 4= 0,085	1= 0,57 2= 0,765 3= 0,135 4= 0,13	1= 0,58 2= 0,775 3= 0,145 4= 0,14	1= 0,58 2= 0,77 3= 0,145 4= 0,135
Compiler	1= 0,445 2= 0,069 3= 0,045 4= 0,04	1= 0,445 2= 0,695 3= 0,045 4= 0,04	1= 0,45 2= 0,695 3= 0,05 4= 0,035	1= 0,445 2= 0,695 3= 0,05 4= 0,04
Omnikon-Basic	1= 0,815	2= 0,495	3= 0,495	4= 0,085

Fehlt nur noch der SAM!! Wer hat Zeit, den Benchmark-Test 1 bis 4 mit dem SAM auszuprobieren??
Danke auch an Uwe Riemer für die Anregung und Austesten mit dem ToBoS-Compiler!
Gut gehackt ins Neue Jahr wünscht Euch Euer

Richard Raddatz, Pfarrgasse 5, 7050 Waiblingen

Sound-Sampling

Sampling nennt man das digitale Abspeichern analoger Signale, wie Töne und Geräusche. Es findet vorwiegend in den Bereichen der Musik und Sprache(rkennung) Anwendung. Ein Computer kann, innerhalb seiner hardwaremäßigen Fähigkeiten, diese Töne manipulieren, z.B. Hüllkurven verändern, Frequenzen unterdrücken etc.
Das Programm "Mini-Studio" von M. Pilia ist ein Sound-Sampler. An Manipulationen gestattet es lediglich eine Rückwärts-Wiedergabe. Durch Verwendung eines Teiles des Sounds kann man, nur als Beispiel, den bekannten "N-N-N-Nineteen"-Effekt erzielen. Folgende Verbindungen sind zur Benutzung nötig:

- an der EAR-Buchse muß die Quelle des zu samplenden Sounds angeschlossen sein; Kassettenrekorder oder Mikrofon
- an der MIC-Buchse sollte ein Kopfhörer, Monitorlautsprecher oder die Stereo-Anlage angeschlossen sein.

Die Ein- und Ausgabe-Routinen sind kaum zwei händevoll Bytes lang und besonders für Demo- und Spieleschreiber interessant.
Es gibt so einiges, was an dem Programm verbessert werden könnte. Beispiele: einstellbare Taktzeiten, Schläge pro Minute, Umorganisation des Speichers - statt die Sounds von 34048 an aufwärts lieber von 64512 an abwärts, um Platz für mehr Bars zu bekommen - etc.
Meine Idee ist folgende: da es, zumindest als PD, meines Wissens noch kein adäquates Programm gibt, sollten sich interessierte User zusammentun und daraus ein Standard-Programm entwickeln. Ich wollte nicht nur meine ganz eigenen Vorstellungen verwirklichen. Deshalb wurde nur der Basic-Teil etwas gestrafft. Sollten dadurch Unklarheiten beim Verstehen des Programmablaufes entstanden sein, so wenden Sie sich an mich. Peter Miosga, Holtbredde 11, 4354 Datteln

OPUS INTERN (2)

Die wichtigsten Bauteile

Einleitung :

1.OPUS-ROM: ein EPROM mit der jeweiligen ROM-Version. Darin ist das Programm der OPUS enthalten, vergleichbar dem Spektrum-ROM
2.OPUS-RAM: Zwei Kilobyte statisches RAM, nicht immer vorhanden. Wird verwendet, um Diskettendaten und Sprungtabellen abzulegen. Ist es nicht vorhanden, können nur Disketten 40-Tracks, einseitig behandelt werden. Das Nachrüsten ist einfach und empfehlenswert.
3.PIO-IC : Dieser IC bedient vor allem den Druckerport. Er wird aber auch für die Diskettensteuerung verwendet.
4.FDC : Der Disk-Controller WD 1770 ist ein IC, welcher ein einzelnes Byte von der Diskette lesen oder schreiben kann. Das Auslesen ganzer Sektoren macht also das Programm, welches den Controller steuert.
5.PIO-IC : Dieser zweite IC ist nicht programmierbar wie der unter Nummer 3, sondern kann fünf Leitungen auf den Datenbus durchschalten. Er wird für den Joystickport verwendet.

Eingemachtes :
Es wäre zuviel, jetzt jedes Teil genau zu beschreiben, da ohnehin jedes Teil nochmal drankommt. Deshalb hier nur Technische Daten.

1. Das OPUS-ROM ist ein 2764, also ein 8 KByte-EProm. Adressiert wird es, wenn im Spektrum die Adressen 08H oder 1708H angesprungen werden, verlassen wird es mit JP 1748H.
2. Der RAM-Baustein ist ein 6116, 2 KByte statisches RAM.
3. Für den Druckerport und Drive Select wird der MC 68A21 verwendet. Das ist ein Baustein vergleichbar der Z80 PIO. 16 Ports, TRI-STATE.
4. Der FDC (Floppy Disk Controller) ist ein Werstern Digital WD 1770. Übertragungsrate 31.000 Byte/sec. Leider durch die OPUS-Software stark gebremst.
5. Der letzte PIO-IC ist ein 74LS385. Der hat TRI-STATE-Ausgänge, und enthält 6 Treiber/Empfänger.
   TRI-STATE heißt, daß die Ports drei Zustände einnehmen können. High, LOW, und hochohmig (offen).

Dieter Hucke, Korbacherstraße 241, 3500 Kassel

Farbige Bilder vom Monitor

Um farbige Bilder von einem Monitor zu bekommen braucht man eine Spiegelreflex-kamera, ein Teleobjektiv (Mindestbrennweite 130, 200 wäre optimal), einen Farb-monitor, eine Wasserwaage und ein Stativ für die Kamera.
Als erstes muß der Monitor mit der Wasserwaage ausgerichtet werden, damit man eine gemeinsame Bildachse zur Kamera bekommt. Die Bildröhre muß natürlich in der Waage sein. Die Kamera wird auf dem Stativ ungefähr 1,5 - 1,8 Meter vor dem Monitor plaziert.

Einstellung der Kamera: Bei einem 21 DIN Film wählen wir Blende 8 - 11 und eine Belichtungszeit von 3 - 4 Sekunden.
Mit dieser Zeit sollte man ein wenig experimentieren.

Ich würde mich freuen, wenn Ihr mal schreiben würdet, wie Eure Fotos geworden sind, und mit welcher Systemzusammenstellung und Einstellung es am besten funktioniert hat.
Mit freundlichen Grüßen

Uwe Kapuschinski, Morgenstr. 35, 4750 Unna, Tel. (02203) 12242

Hisoft Pascal auf dem Speccy
Teil 2

So, nachdem im letzten Teil unser erstes Programm das Laufen lernte - oder auch nicht - muß ich einiges nachtragen, was ich total vergaß. Manche werden beim Eingeben der Befehle vielleicht mal ein Semikolon vergessen haben und beim Compilieren gab's haufenweise Errors.

Da also nicht jeder so perfekt sein kann wie ich (Lüge!), hier die lebenswichtigen Befehle des Befehlsmodus (!).

CAPS - 1	Verlassen des Editors in den Befehlsmodus
C a	Programm wird ab Zeile a compiliert, wird a nicht angegeben geschieht dies von der ersten Zeile an
D a,b	Zeilen von a bis b werden gelöscht (für immer!)
E a	holt Zeile a in den Zeileneditor (-> Zeileneditor)
G,,string	lädt ein Programm namens string
I a,b	Springt in den Editor ab Zeile a in b-Schritten
K a	gibt an, wieviel Zeilen mit List (-> L) gelistet werden, bevor der Bildschirm hochgeschoben wird
L a,b	listet das Programm von Zeile a bis b
N a,b	Remumberfunktion, Programm wird von Zeile a an in b-Schritten neu nummeriert
P a,b,string	Programm wird von Zeile a - b unter dem Namen string abgespeichert
R	startet ein Programm
T	Programm wird compiliert und abgespeichert

Und hier nun die Befehle des Zeileneditors, aber vorher muß ich dazu einiges erläutern. Holt man eine Zeile in den Zeileneditor, so erscheint erst einmal die Zeile komplett und darunter nochmal die Zeilennummer, dahinter ein C-Cursor. Nun drückt man solange SPACE, bis die Zeile einschließlich des Fehlers erschienen ist, drückt dann I, worauf ein Sternchen als Cursor erscheint, und drückt DELETE und danach zweimal ENTER.

Einfacher geht es, wenn man bis eine Stelle vor dem Fehler SPACE drückt und dann K, gefolgt von zweimal ENTER.

Beispiele: 10 PROGRAM DIES_IST_EIN_TEST (_ sind im Programmnamen verboten!)
10 C (=Cursor)

Wir drücken SPACE bis "...DIES", dann einmal K. K steht für KILL und killt sozusagen das nächste Zeichen. So löschen wir jetzt jedes _-Zeichen. Jetzt fehlt unserer Fehlerzeile noch ein Semikolon. Hierfür verwenden wir das I, für Insert. Am Ende unserer Zeile drücken wir also I, das Sternchen erscheint, wir machen unser Semikolon, beenden das Insert mit ENTER und verlassen den Zeileneditor mit ENTER. Hier wiederum lebenswichtige Befehle:

SPACE	Um ein Zeichen weiter
I	Einfügen von Zeichen, aber auch Löschen mit DELETE
K	Löschen des nächsten Zeichens
Q	Verlassen des Zeileneditors ohne Änderungen
R	Die Zeile wird nochmal ohne Änderungen in den Zeileneditor geladen
X	springt zum Zeilenende
Z	löscht die ganze Zeile
CAPS - 0	Ein Zeichen zurück (kein DELETE!)
CAPS - 5	löscht Zeile mitsamt Zeilennummer
ENTER	verläßt den Zeileneditor

So, dies einmal zur Handhabung des Befehlsmodus und des Zeileneditors.
Übrigens, wenn jemand sein Programm ausgedruckt haben will, so muß er am Anfang der ersten Zeile **{$P}** einfügen. Wie man etwas einfügt, wissen wir ja jetzt. Wird das Programm nun compiliert, wird das Programm mit Adressen anstatt auf den Bildschirm in den Druckerkanal geschickt.

Jetzt aber mal zu richtigen Pascal-Befehlen. Unser erstes Programm konnte ein phänomenales HALLO auf den Bildschirm ausgeben, aber nur einmal. Was ist aber, wenn wir zehnmal HALLO haben wollen? - Wer hat da "10 * R drücken" gesagt?! Auch eine Lösung, aber ziemlich arbeitsintensiv und unzuverlässig (was ist, wenn einer nicht bis 10 zählen kann, hä?!).
In BASIC haben wir hierfür unsere For-Next-Schleife. Wäre doch schön, wenn es sowas auch in Pascal gibt - gibt's! (Hallo Günter! Der, den ich meine, versteht das schon. Sch...Internwitz).
In Pascal unterscheidet sich aber der Syntax und die Handhabung etwas von Basic. Der Syntax in Pascal lautet:

```
For i:= anfang TO ende DO
Begin (bei mehr als einem Befehl)
: Befehlsfolge
End; (bei mehr als einem Befehl)
```

Also kein Next, sondern ein End, Semikolon nicht vergessen! Wenn man in negativen Schritten zählen will, lautet der Syntax so:

```
For i:= ende Downto anfang Do usw.
```

Jetzt ist es bei Pascal so, daß man nicht einfach wild Variablen benutzen darf, wie man will. Sie müssen vorher vereinbart werden. Das geschieht nach der PROGRAM-Zeile.
In unserem Fall nehmen wir mal an, Anfang und Ende sind Zahlen, 1 und 10. Wir haben also eine Variable, i. Wir wollen ja in Ganzzahlschritten zählen, 1, 2, 3, ..., 10. Das müssen wir auch vereinbaren.

Der Syntax einer Vereinbarung sieht so aus:

```
VAR variable:typ;
```

In unserem Fall ist die Variable i und der Typ ist Ganzzahl, was in Pascal mit INTEGER ausgedrückt wird. Wir schreiben also:

```
VAR i:INTEGER;
```

So, und nun fügt die For-End-Schleife in euer erstes Programm ein. Und wenn es klappt, freut euch!
Huh, jetzt habe ich schon wieder so viel geschrieben, wo soll das hinpassen? Jetzt noch einige Anfragen in eigener Sache:
Aufruf an alle GFX-(Grafik-) Künstler am Speccy. Durchforstet eure Screensammlung. Ich bräuchte Screens von Landschaften (Wüsten, Steppen, Wälder, Gebirge, Ruinen von futuristischen Städten), von Aliens (schön groß und häßlich), von Planetensystemen und Raumschiffen. Wer sowas hat, schickt es mir, je mehr desto besser.
Mein zweites Problem: Ich habe Beta- und Opus-Disk und möchte beide gleichzeitig am Speccy haben um Backups von Beta auf Opus und umgekehrt zu machen. Wer kann mir helfen?
Außerdem möchte ich Speccy + Peripherie in ein Gehäuse bauen, dazu fliegt der Trafo der Opus raus, ich benutze ein Xerox Netzteil. Ich brauche detaillierte Anschlußpläne der Stromversorgung der Opus, da ich 1. kein Meßgerät habe (peinlich) und 2. die Kennummer des Trafos unkenntlich gemacht wurde.
Wer von den anderen zwei Amiga-Usern im SPC hat seinen Speccy mit dem Amiga gekoppelt - problemlos? Ich hab' da seit letzter Zeit erheblich Probleme (via RS 232 mit ZX Telecom und Amiga DFÜ).
Und last but not least: Hat jemand im Opus-ROM eine graphische Benutzeroberfläche eingebaut, ähnlich wie beim Beta das Vision? Wenn ja, bitte melden - Dringend!

Lord Luxor, Wieblinger Weg 55, 6900 Heidelberg



Tips und Tricks

Heute ein Beitrag für Tips und Tricks zur Verlängerung der Lebensdauer des Spectrums.
Dabei geht es um die Abführung der Stauwärme im Spectrum. Ein einfacher Weg, dies zu erreichen, ist in der nebenstehenden Abbildung erklärt. Je kälter das Gerät, desto länger die Lebensdauer.

Dieter Konietzko, Ostring 6, 2393 Sörup

Screen-Manipulationen

Hier noch eine letzte Screen-Routine zur Umsetzung in MC. Diesmal geht es um das Verkleinern:

```
- FOR x=1 TO 6144: POKE 50000+x,PEEK (16384+x): NEXT x: LET u=56912: LET
v=50000: FOR l=1 TO 3072: LET c=PEEK v: LET d=PEEK (v+1): LET e=128: LET f=192:
LET g=0: FOR i=1 TO 4: IF c>=f THEN LET c=c-f: LET g=g+e
- LET e=e/2: LET f=f/4: NEXT i: LET f=192: FOR i=1 TO 4: IF d>=f THEN LET d=d-f:
LET g=g+e
- LET e=e/2: LET f=f/4: NEXT i: POKE u,c: LET u=u+1: LET v=v+2: NEXT l
```

Patrick Thiel, Königsberger Str. 11, 4796 Salzkotten, Tel. 05258/5197

Anzeigen

Brauche DRINGEND 48er oder -Platine ! (evtl. defekt)
Frank Meurer, Schulstr. 21, 5047 Wesseling, Tel. 02236/46966

Ernsthaftes Angebot: Tausche Atari 1040 STF mit Monochrom-Monitor, Maus, Omikron-Basic, GFA-Basic, Protext 2.1 und div. PD-Software gegen einen SAM-Coupe mit Laufwerk!!!
Wolfgang Haller, Ernastr. 33, 5000 Köln 80, Tel. 0221/685946

Suche Kontakte zu Multiface 3 Usern. Wer kennt sich aus? Ich habe einen 128+2 (only Tape). Suche auch Soft. Bernd Karle, Kleinfeldele 36, 7840 Müllheim

---

Verkaufe:
- ZX Spectrum 48K eingebaut in VOBIS PC Gehäuse, Monitor/Fernseherausgang, Reset-Taste, abgesetzte PC-Tastatur von Phillips (Kabellänge ca. 1m, gepuffert), Joystickanschlüsse für Sinclair L/R, Rank Xerox Netzteil, läuft total absturzfrei, für 170 DM.
- Original Softwarepaket (ca. 35 Cassetten, u.a. Pud Pud, Bruce Lee, The Legend of Avalon, neue Crash-Tapes, Commando, JSW): 60 DM. - Spectrum Platine (3B) zum Anschl. incl. ULA/CPU/RAMs: 40 DM. - Dk'tronics progr. Joystick-IF, durchgeführter Bus: 30 DM. - Beta Disk System 4.12 mit Vision Benutzeroberfläche, 3,5" EPSON Laufwerk (auch für z.B. Atari ST) DSDD, 100 Disketten: 350 DM. - Sinclair QL incl. Mouse für 130 DM. - Neuer Grundig Datenrekorder mit Garantie: 35 DM. - Z80 Buch (R.Zaks/Sybex): 20 DM, ROM Disassembly (Melbourne): 15 DM. - SAM Coupe Technical Manual V.2.2: 5 DM, V.2.3: 10 DM. - RAM-Turbo Joyst.-IF, durchgef. Bus (Sinclair/Kempston/Protek): 30 DM. - Stoneship Tastatur für 48K mit Beepverstärker + Sinclair Joyst.: 30 DM. - Verschiedene Bücher: je 3 bzw. 5 DM.

Bernhard Lutz, Obermühlstr. 24 (bei Wünschel), 6729 Bellheim, Tel. (07272) 6868